Substitute Specification (clean version without markings)

FEATURE WEIGHTING IN K-MEANS CLUSTERING

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention generally relates to data clustering and in particular, concerns a method and system for providing a framework for integrating multiple, heterogeneous feature spaces in a *k*-means clustering algorithm.

Description of the Related Art

Clustering, the grouping together of similar data points in a data set, is a widely used procedure for analyzing data for data mining applications. Such applications of clustering include unsupervised classification and taxonomy generation, nearest-neighbor searching, scientific discovery, vector quantization, text analysis and navigation, data reduction and summarization, supermarket database analysis, customer/market segmentation, and time series analysis.

One of the more popular techniques for clustering data of a set of data records includes partitioning operations (also referred to as finding pattern vectors) of the data using a k-means

clustering algorithm which generates a minimum variance grouping of data by minimizing the sum of squared Euclidean distances from cluster centroids. The popularity of the k-means clustering algorithm is based on its ease of interpretation, simplicity of use, scalability, speed of convergence, parallelizability, adaptability to sparse data, and ease of out-of-core use.

The k-means clustering algorithm functions to reduce data. Initial cluster centers are chosen arbitrarily. Records from the database are then distributed among the chosen cluster domains based on minimum distances. After records are distributed, the cluster centers are updated to reflect the means of all the records in the respective cluster domains. This process is iterated so long as the cluster centers continue to move and converge and remain static. Performance of this algorithm is influenced by the number and location of the initial cluster centers, and by the order in which pattern samples are passed through the program.

Initial use of the k-means clustering algorithm typically requires a user or an external algorithm to define the number of clusters. Second, all the data points within the data set are loaded into the function. Preferably, the data points are indexed according to a numeric field value and a record number. Third, a cluster center is initialized for each of the predefined number of clusters. Each cluster center contains a random normalized value for each field within the cluster. Thus, initial centers are typically randomly defined. Alternatively, initial cluster center values may be predetermined based on equal divisions of the range within a field. In a fourth step, a routine is performed for each of the records in the database. For each record number from one to the current record number, the cluster center closest to the current record is determined. The record is then assigned to that closest cluster by adding the record number to the list of records previously assigned to the cluster. In a fifth step, after all of the records have been assigned to a cluster, the cluster center for each cluster is adjusted to reflect the averages of data

values contained in the records assigned to the cluster. The steps of assigning records to clusters and then adjusting the cluster centers is repeated until the cluster centers move less than a predetermined epsilon value. At this point the cluster centers are viewed as being static.

A fundamental starting point for machine learning, multivariate statistics, or data mining, is where a data record can be represented as a high-dimensional feature vector. In many traditional applications, all of the features are essentially of the same type. However, many emerging data sets are often have many different feature spaces, for example:

- Image indexing and searching systems use at least four different types of features: color, texture, shape, and location.
- Hypertext documents contain at least three different types of features: the words, the out-links, and the in-links.
- XML has become a standard way to represent data records; such records may have a number of different textual, referential, graphical, numerical, and categorical features.
- Profile of a typical on-line customer such as an Amazon.com customer may contain purchased books, music, DVD/video, software, toys, etc. These above examples illustrate that data sets with multiple, heterogeneous features are indeed natural and common. In addition, many data sets on the University of California Irvine Machine Learning and Knowledge Discovery and Data Mining repositories contain data records with heterogeneous features. Data clustering is an unsupervised learning operation whose output provides fundamental techniques in machine learning and statistics. Statistical and computational issues associated with the k-means clustering algorithm have extensively been used for these clustering operations. The same cannot be said, however, for another key ingredient for multidimensional data analysis: clustering data records having multiple, heterogeneous feature spaces.

SUMMARY OF THE INVENTION

The invention provides a method and system for integrating multiple, heterogeneous feature spaces in a *k*-means clustering algorithm. The method of the invention adaptively selects the relative weights assigned to various features spaces, which simultaneously attains good separation along all of the feature spaces.

The invention integrates multiple feature spaces in a *k*-means clustering algorithm by assigning different relative weights to these various features spaces. Optimal feature weights are also determined that can be incorporated with this algorithm that lead to a clustering that simultaneously minimizes the average intra-cluster dispersion and maximizes the average inter-cluster dispersion along all the feature spaces.

DESCRIPTION OF THE DRAWINGS

The foregoing will be better understood from the following detailed description of preferred embodiments of the invention with reference to the drawings, in which:

FIGs. 1A and 1B show a data computing system and method of the invention respectively;

FIGs. 2A, 2B, 2C, and 2D show graphical results of an example using the invention;

FIG. 3 shows the feasible weights for the second exemplary use of the invention wherein when the feature space is 3, and the triangular region formed by the intersection of the plane at $\alpha_1 + \alpha_2 + \alpha_3 = 1$ with the nonnegative orthant of $\mathbb{R}^3$;

FIG. 4 shows a newsgroups data set, in which plot macro-p versus the objective function

$Q_1 \times Q_2 \times Q_3$ for various different weight tuples; and

FIG. 5 is a flowchart illustrating a preferred method of the invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS OF THE INVENTION

1. Introduction

While the invention is primarily disclosed as a method, it will be understood by a person of ordinary skill in the art that an apparatus, such as a conventional data processor, including a CPU, memory, I/O, program storage, a connecting bus, and other appropriate components, could be programmed or otherwise designed to facilitate the practice of the method of the invention. Such a processor would include appropriate program means for executing the method of the invention. Also, an article of manufacture, such as a pre-recorded disk or other similar computer program product, for use with a data processing system, could include a storage medium and program means recorded thereon for directing the data processing system to facilitate the practice of the method of the invention. It will be understood that such apparatus and articles of manufacture also fall within the spirit and scope of the invention.

FIG. 1A shows an exemplary data processing system for practicing the disclosed feature weighted K-means data clustering analysis methodology that includes a computing device in the form of a conventional computer 20, including one or more processing units 21, a system memory 22, and a system bus 23 that couples various system components including the system memory to the processing unit 21. The system bus 23 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures.

The system memory includes read only memory (ROM) 24 and random access memory (RAM) 25. A basic input/output system 26 (BIOS), containing the basic routines that helps to transfer information between elements within the computer 20, such as during start-up, is stored in ROM 24.

The computer 20 further includes a hard disk drive 27 for reading from and writing to a hard disk, not shown, a magnetic disk drive 28 for reading from or writing to a removable magnetic disk 29, and an optical disk drive 30 for reading from or writing to a removable optical disk 31 such as a CD-ROM or other optical media. The hard disk drive 27, magnetic disk drive 28, and optical disk drive 30 are connected to the system bus 23 by a hard disk drive interface 32, a magnetic disk drive interface 33, and an optical drive interface 34, respectively. The drives and their associated computer-readable media provide nonvolatile storage of computer readable instructions, data structures, program modules and other data for the computer 20. Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 29, and a removable optical disk 31, it should be appreciated by those skilled in the art that other types of computer readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, random access memories (RAMs), read only memories (ROMs), and the like, may also be used in the exemplary operating environment. Data and program instructions can be in the storage area that is readable by a machine, and that tangibly embodies a program of instructions executable by the machine for performing the method of the present invention described herein for data mining applications.

A number of program modules may be stored on the hard disk, magnetic disk 29, optical disk 31, ROM 24 or RAM 25, including an operating system 35, one or more application

programs 36, other program modules 37, and program data 38. A user may enter commands and information into the computer 20 through input devices such as a keyboard 40 and pointing device 42. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 21 through a serial port interface 46 that is coupled to the system bus, but may be connected by other interfaces, such as a parallel port, game port or a universal serial bus (USB). A monitor 47 or other type of display device is also connected to the system bus 23 via an interface, such as a video adapter 48. In addition to the monitor, personal computers typically include other peripheral output devices (not shown), such as speakers and printers. The computer 20 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 49. The remote computer 49 may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer 20, although only a memory storage device 50 has been illustrated in FIG. 1A. The logical connections depicted in FIG. 1A include a local area network (LAN) 51 and a wide area network (WAN) 52. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

When used in a LAN networking environment, the computer 20 is connected to the local network 51 through a network interface or adapter 53. When used in a WAN networking environment, the computer 20 typically includes a modem 54 or other means for establishing communications over the wide area network 52, such as the Internet. The modem 54, which may be internal or external, is connected to the system bus 23 via the serial port interface 46. In a networked environment, program modules depicted relative to the computer 20, or portions

thereof, may be stored in the remote memory storage device. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

The method of the invention as shown in general form in FIG. 1B, may be implemented using standard programming and/or engineering techniques using computer programming software, firmware, hardware or any combination or sub-combination thereof. Any such resulting program(s), having computer readable program code means, may be embodied or provided within one or more computer readable or usable media such as fixed (hard) drives, disk, diskettes, optical disks, magnetic tape, semiconductor memories such as read-only memory (ROM), etc., or any transmitting/receiving medium such as the Internet or other communication network or link, thereby making a computer program product, i.e., an article of manufacture, according to the invention. The article of manufacture containing the computer programming code may be made and/or used by executing the code directly from one medium, by copying the code from one medium to another medium, or by transmitting the code over a network.

The computing system for implementing the method of the invention can be in the form of software, firmware, hardware or any combination or sub-combination thereof, which embody the invention. One skilled in the art of computer science will easily be able to combine the software created as described with appropriate general purpose or special purpose computer hardware to create a computer system and/or computer subcomponents embodying the invention and to create a computer system and/or computer subcomponents for carrying out the method of the invention.

The method of the invention is for clustering data, by establishing a starting point at step 1 as shown in FIG. 1B wherein, a given data set having m feature spaces, and each data object

(record) is represented as a tuple of *m* feature vectors. To cluster, a measure of distortion between two data records is needed. Since, different types of features may have radically different statistical distributions, in general, it is unnatural to disregard fundamental differences between various different types of features and to impose a uniform, un-weighted distortion measure across disparate feature spaces. In Section 2 below, a distortion between two data records as a weighted sum of suitable distortion measures on individual component feature vectors is provided; where the distortions on individual components are allowed to be different. In Section 3 below, using a convex programming formulation, the classical Euclidean *k*-means algorithm is generalized to use the weighted distortion measure. In Section 4 below, optimal feature weights are selected that lead to a clustering that simultaneously minimizes the average intra-cluster dispersion and maximizes the average inter-cluster dispersion along all the feature spaces. In Section 5, an outline evaluation strategy is provided. In Sections 6 and 7, two exemplary uses of the invention are provided for a) clustering data sets with numerical and categorical features; and b) clustering text data sets with words, 2-phrases, and 3-phrases respectively. Using data sets with a known ground truth classification, the clusterings are empirically demonstrated that correspond to the optimal feature weights and deliver nearly optimal precision/recall performance.

Feature weighting may be thought of as a generalization of feature selection where the latter restricts attention to weights that are either 1 (retain the feature) or 0 (eliminate the feature), see Wettschereck et al., *Artificial Intelligence Review* in the article entitled "A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms," Vol. 11, pps. 273-314, 1997. Feature selection in the context of supervised learning has a long history in machine learning, see, for example, Blum et al., *Artificial Intelligence*, "Selection of

relevant features and examples in machine learning," Vol. 97, pps. 245-271, 1997. Feature selection in the context of unsupervised learning has only recently been systematically studied.

2. Data Model and a Distortion Measure

2.1 Data Model: Assume that a set of data records where each object is a tuple of m component feature vectors are given. A typical data object is written as: $x = (F_1, F_2, \ldots F_m)$, where the l-th component feature vector F_l, $1 \le l \le m$, is to be thought of as a column vector and lies in some (abstract) feature space F_l. The data object x lies on the m-fold product feature space $F = F_1 \times F_2 \times \ldots F_m$. The feature spaces $\{F_l\}_{l=1}^{m}$ can be dimensionally different and possess different topologies, hence, the data model accommodates heterogeneous types of features. There are two examples of feature spaces that include:

Euclidean Case: F_l is either $\mathbb{R}^{f_l}$, $f_l \ge 1$, or some compact submanifold thereof.

Spherical Case: F_l is the intersection of the f_l-dimensional, $f_l \ge 1$, unit sphere with the non-negative orthant of $\mathbb{R}^{f_l}$.

2.2 A Weighted Distortion Measure: Measuring distortion between two given two data records $x = (F_1, F_2, \ldots F_m)$ and $\tilde{x} = \left(\tilde{F}_1, \tilde{F}_2, \ldots \tilde{F}_m\right)$. For $1 \le l \le m$, let D_l denote a distortion measure between the corresponding component feature vectors F_l and $\tilde{F}_l$. Mathematically, only two needed properties of the distortion function:

- $D_l: F_l \times F_l \to (0, \infty)$
- For a fixed $\tilde{F}_l$, D_l is convex in F_l.

Euclidean Case The squared-Euclidean distance:

$$D_l\left(F_l \tilde{F}_l\right) = \left(F_l - \tilde{F}_l\right)^T \left(F_l - \tilde{F}_l\right)$$

trivially satisfies the non-negativity and, for $\lambda \in [0,1]$, the convexity follows from :

$$D_l\left(F_l, \lambda \tilde{F}_l' + (1-\lambda)\tilde{F}_l''\right) \leq \lambda D_l\left(F_l, \tilde{F}_l'\right) + (1-\lambda) D_l\left(F_l, \tilde{F}_l''\right).$$

Spherical Case The cosine distance $D_l\left(F_l, \tilde{F}_l\right) = 1 - F_l^T \tilde{F}_l$ trivially satisfies the non-negativity and, for $\lambda \in [0,1]$, the convexity follows from:

$$D_l\left(F_l, \frac{\lambda \tilde{F}_l' + (1-\lambda)\tilde{F}_l''}{\left\|\lambda \tilde{F}_l' + (1-\lambda)\tilde{F}_l''\right\|}\right) \leq \lambda D_l\left(F_l, \tilde{F}_l'\right) + (1-\lambda) D_l\left(F_l, \tilde{F}_l''\right),$$

where $||\ ...||$ denotes the Euclidean-norm. The division by: $||\ \lambda \tilde{F}_l' + (1-\lambda)\tilde{F}_l''\ ||$ ensures that the second argument of D_l is a unit vector. Geometrically, the convexity along the geodesic are defined as connecting the two unit vectors $\tilde{F}_l'$ and $\tilde{F}_l''$ and not along the chord connecting the two. Given m valid distortion measures $\{D_l\}_{l=1}^m$ between the corresponding m component

feature vectors of x and $\tilde{x}$, a weighted distortion measure between x and $\tilde{x}$ is defined as:

$$D^{\alpha}\left(x,\tilde{x}\right) = \sum_{l=1}^{m} \alpha_l D_l\left(F_l, \tilde{F}_l\right),$$

where the feature weights $\{\alpha_l\}_{l-1}^{m}$ are non-negative and sum to 1 and $\alpha = (\alpha_l, \alpha_2, \ldots \alpha_m)$.

The weighted distortion D^{α} is a convex combination of convex distortion measures, and hence, for a fixed x, D^{α} is the convex in $\tilde{x}$. The feature weights $\{\alpha_l\}_{l=1}^{m}$ are enabled in the method, and are used to assign different relative importance to component feature vectors. In Section 4 below, appropriate choice of these parameters is made.

3. *k*- Means with Weighted Distortion

3.1. The Problem: Suppose that n-data records are given such that

$$x_i = \left(F_{(i,1)}, F_{(i,2)}, \ldots F_{(i,m)}\right), 1 \le i \le n,$$

where the l-th, $1 \le l \le m$, component feature vector of every data record is in the feature space F_l. Partitioning of the data set $\{x_l\}_{i=1}^{n}$ is sought into *k-disjoint* clusters $\{\pi_u\}_{u=1}^{k}$.

3.2 Generalized Centroids: Given a partitioning $\{\pi_u\}_{u=1}^{k}$, for each partition π_u, write the corresponding generalized centroid as

$$c_u = \left(c_{(u,1)}, c_{(u,2)}, \ldots, c_{(u,m)}\right),$$

where, for $1 < l > m$, the l-th component $c_{(u,l)}$ is in F_l. c_u as the solution of the following convex programming problem is defined as:

$$c_u = \underset{\tilde{x} \in f}{arg\ min} \left(\sum_{x \in \pi_u} D^{\alpha}(x, \tilde{x}) \right). \quad (1)$$

In an empirical average sense, the generalized centroid may be thought of as being the closest in D^{α} to all the data records in the cluster π_u. The key to solving (1) is to observe that D^{α} is component-wise-convex, and, hence, equation (1)can be solved by separately solving for each of its m components $c_{(u,l)}$, $1 < l < m$. In other words, the following m convex programming problem is solved:

$$c_{(u,l)} = \underset{\tilde{F}_l \in F_l}{arg\ min} \left(\sum_{x \in \pi} D_l(F_l, \tilde{F}_l) \right). \quad (2)$$

For the two feature spaces of interest (others as well), the solution of equation (2) can be written in a closed form using a Euclidean and Spherical case, respectively:

$$c_{(u,l)} = \begin{cases} \frac{1}{\Sigma_{Num}} \Sigma_{Num} F_l \\ \frac{\Sigma_{Num} F_l}{\| \Sigma_{Num} F_l \|} \end{cases}$$

where $x = (F_1, F_2 \ldots, F_m)$

3.3 The Method: Referring to FIG. 1B, the method of the invention uses the formulation of equation (1) using the steps below, wherein the distortion is measured of each individual cluster $\pi_u, 1 < u < k$, as:

$$\sum_{x \in \pi_u} D^{\alpha}(x, c_u),$$

and the quality of the entire partitioning $\{\pi_u\}_{u=1}^{k}$ as the combined distortion of all the k clusters: $\sum_{u=1}^{k} \sum_{x \in \pi_u} D^{\alpha}(x, c_u)$. What is sought is k-disjoint clusters such that equation (3) as follows is minimized wherein these k-disjoint clusters are:

$$\pi_1^{\dagger}, \pi_2^{\dagger}, \ldots, \pi_k^{\dagger}, \text{and}$$

$$\left\{ \pi_u^{\dagger} \right\}_{u=1}^{k} = \underset{\{\pi_u\}_{u=1}^{k}}{arg\,min} \left(\sum_{u=1}^{k} \sum_{x \in \pi_u} D^{\alpha}(x, c_u) \right), \tag{3}$$

where the feature weights $\alpha = (\alpha_1, \alpha_2, \ldots \alpha_m)$ are fixed. When only one of the weights $\{\alpha_l\}_{l=1}^{m}$ is nonzero, the maximization problem (3) is known to be NP-complete, meaning no known algorithm exists for solving the problem in polynomial time. *K*-means is used, which is an efficient and effective data clustering algorithm. Moreover, *k*-means can be thought of as a gradient ascent method, and, hence, never increases the objective function and eventually converges to a local minima.

In FIG. 1B, an overview of the processing components is shown for clustering data. These processing components perform data clustering on data records stored on a storage medium such as the computer system's hard disk drive 27. The data records typically are made up of a number of data fields or attributes. Examples of such data records are discussed below in the two examples implementing the invention.

The components that perform the clustering require three inputs: the number of clusters K, a set of K initial starting points, and the data records to be clustered. The clustering of data by these components produces a final solution as shown in step 5 as an output. Each of the K clusters of this final solution is represented by its mean (centroid) where each mean has d components equal to the number of attributes of the data records and a fixed feature weight of the m-feature spaces.

A refinement of feature weights in step 4 below produces better clustering from the data records to be clustered using the methodology of the invention. A most favorable refined starting point produced using a good approximation of an initial starting point is discussed below that would move the set of starting points that are closer to the modes of the data distribution.

At Step 1: An initial point with an arbitrary partitioning of the data records of the data records to be evaluated is provided, wherein, $\left\{\pi_u^{(o)}\right\}_{u=1}^{k}$. Let $\left\{c_u^{(o)}\right\}_{u=1}^{k}$ denote the generalized centroids associated with the given partitioning. Set the index of iteration $t = 0$. A choice of the initial partitioning is quite crucial to finding a good local minima; to achieve this, see a method for doing this technique as taught in U.S. patent 6,115,708 hereby incorporated by reference.

At Step 2: For each data record $x_i, 1 \le i \le n$, find the generalized centroid that is closest